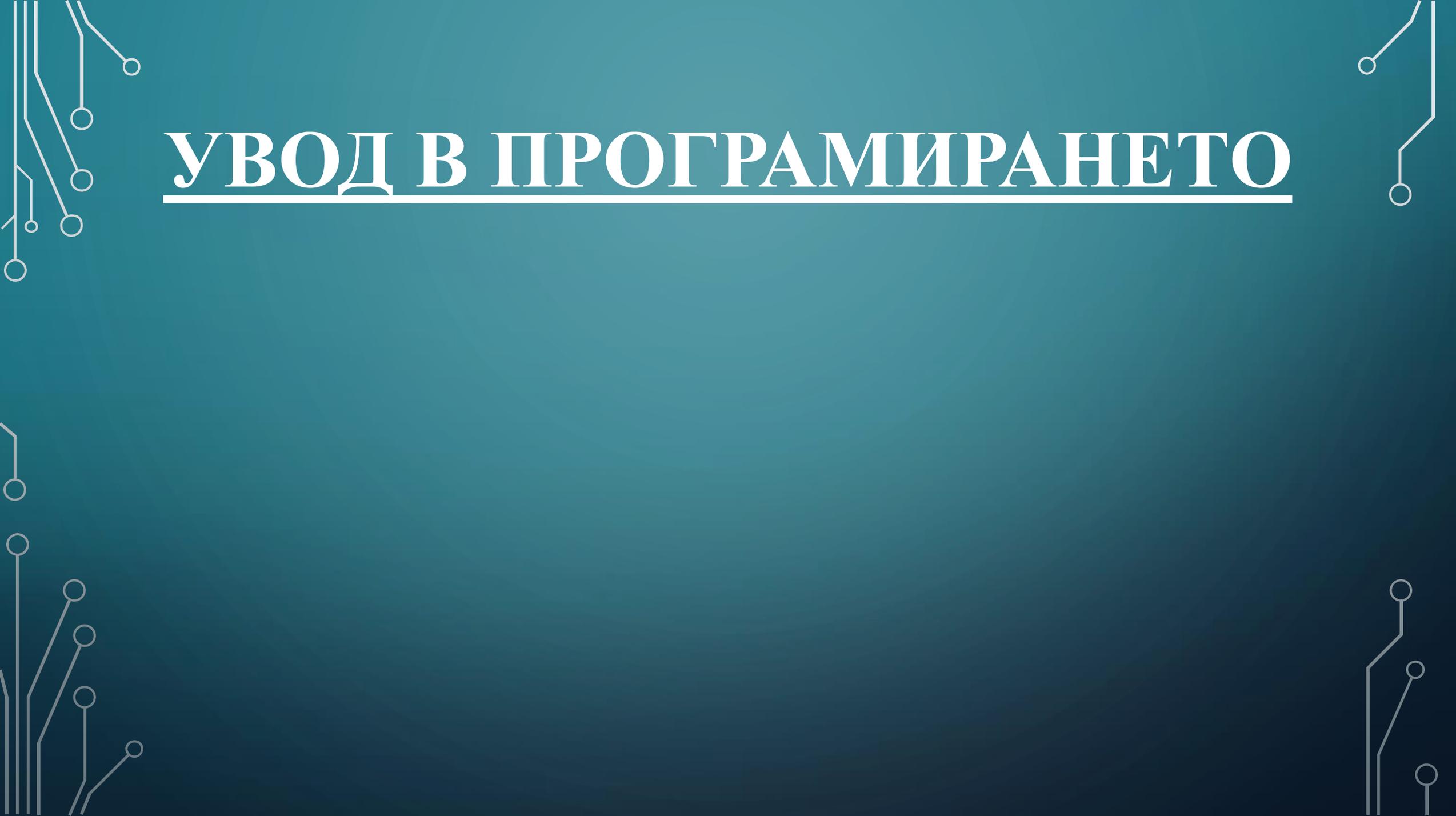


УВОД В ПРОГРАМИРАНЕТО

The background is a dark teal color with white decorative elements resembling circuit board traces. These traces are located in the four corners of the page, forming a frame around the central text. The traces consist of thin white lines that branch out and terminate in small white circles, mimicking the look of electronic components or connections on a PCB.

СЪДЪРЖАНИЕ

- Програмиране(основни понятия)
- Основни понятия от компютърната архитектура
- Представяне на информацията
- Алгоритми и средства за описанието им
- Езици за програмиране (основни елементи на C++)

ДЕФИНИЦИЯ

- Процесът на проектиране и реализиране на програми се нарича **компютърно програмиране**
- **Програма** е списък от инструкции, които описват действията , които трябва да бъдат извършени от компютъра

ПОГЛЕД КЪМ КОМПЮТЪРА

- Средство за представяне и обработване на информация
- Може да изпълнява определена последователност от операции
- Числови изчисления, аналитични преобразувания, обработка на данни, входно/изходни операции
- Хардуер-Апаратно осигуряване
- Софтуер-Програмно осигуряване

ХАРДУЕР

- Централен процесор (CPU) |
- Оперативна памет (Random Access Memory- RAM/Read Only Memory- ROM)
- Външна памет (Hard disk/CD/DVD/Flash)
- Периферни устройства(Клавиатура/Мишка/Монитор)
- Дънна платка(шина)

ПАМЕТ И ПРЕДСТАВЯНЕ НА ИНФОРМАЦИЯТА

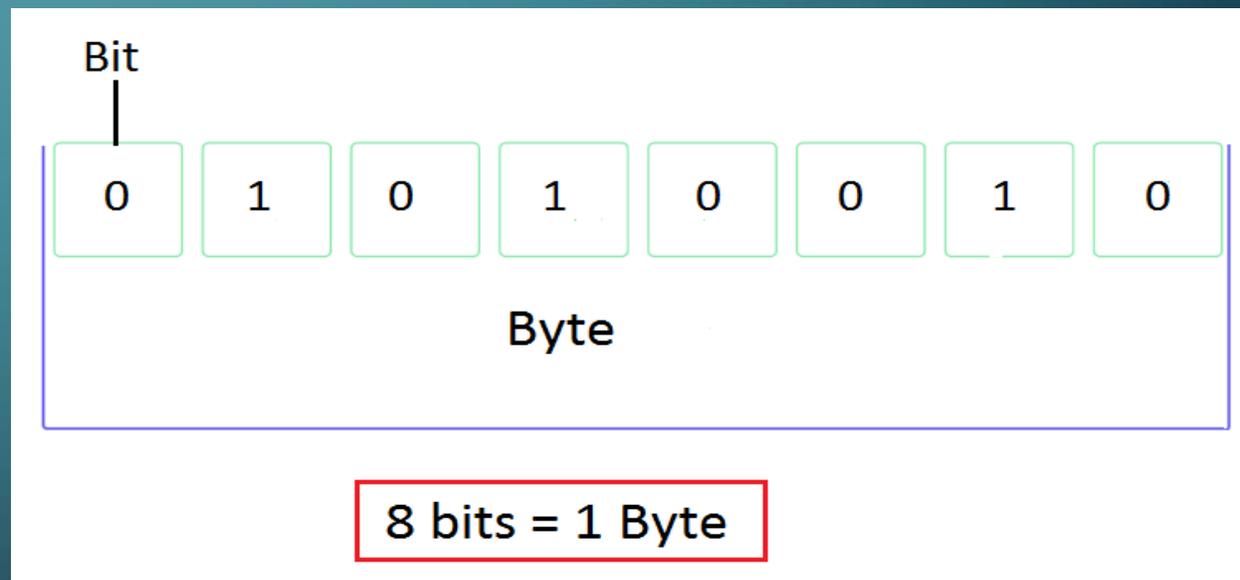
- **Информация подлежаща на обработка, представлява абстракция на някакъв фрагмент от действителния свят**
- Двоична бройна система
- Битове и байтове

ПАМЕТ

```
11111001101110111100011110111010111010001111111011111110
1100110111111011000100100000010111011101110111111111111
11111000111111000111111111111110011110011111100101111111
0111111011001110111110111111101011111011111111101100111010
1111111110001111111001010010100011111011110110100111101
0001111111111011000101000011110010000000111100100100011
101111100111111111010101111111000101110111000001001111110
011100001111011101111100111111001111111001100000110111
1011101101000010011001100011101110000110010000001100111
1101100110001010111101101000111110100011010111110010111
1111111110011000111100111101010000110100111000100100010
1110100001011100111111100000111110111111011110101111100
111001001101011111111110011111111111111111111111001010111111
```

ПАМЕТ

- Изобразена като редица от елементи, всеки от които е носител на информация- 0 и 1 (Бит/**binary digit**)
- **Байт/Byte**- 8 битова информационна единица
- Дума-група от битове (8,16,32,64....)



ПАМЕТ

- Пряк достъп до всяка дума
- Всяка дума се свързва с пореден номер наречен адрес
- Размерът в битовете на думата е свързан с разредността на компютъра(32 бита, 64 бита)

Address	Data
0	00101110
1	01101101
2	00110010
3	00110101
4	11000111
5	10000110
6	01011101
7	00110010
8	10101011
9	00011101
10	01110011
11	00110010
12	01011010
13	11111110

ПАМЕТ

- Най-малкото количество информация – **бит**
- Най-малката адресируема единица от паметта на компютъра е **байт**

	Approx. Bytes	Actual Bytes	Approx. Bits	Typical file/media
1B	1	1	8	Text email, SMS
1KB	$1000\text{B} = 10^3$	$1024\text{B} = 2^{10}$	8×10^3	Word document
1MB	$1000\text{KB} = 10^6$	$1024\text{KB} = 2^{20}$	8×10^6	Digital photo
1GB	$1000\text{MB} = 10^9$	$1024\text{MB} = 2^{30}$	8×10^9	DVD
1TB	$1000\text{GB} = 10^{12}$	$1024\text{GB} = 2^{40}$	8×10^{12}	Hard disk
1PB	$1000\text{TB} = 10^{15}$	$1024\text{TB} = 2^{50}$	8×10^{15}	Cloud?

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

АЛГОРИТМИ И СРЕДСТВА ЗА ОПИСАНИЕТО ИМ

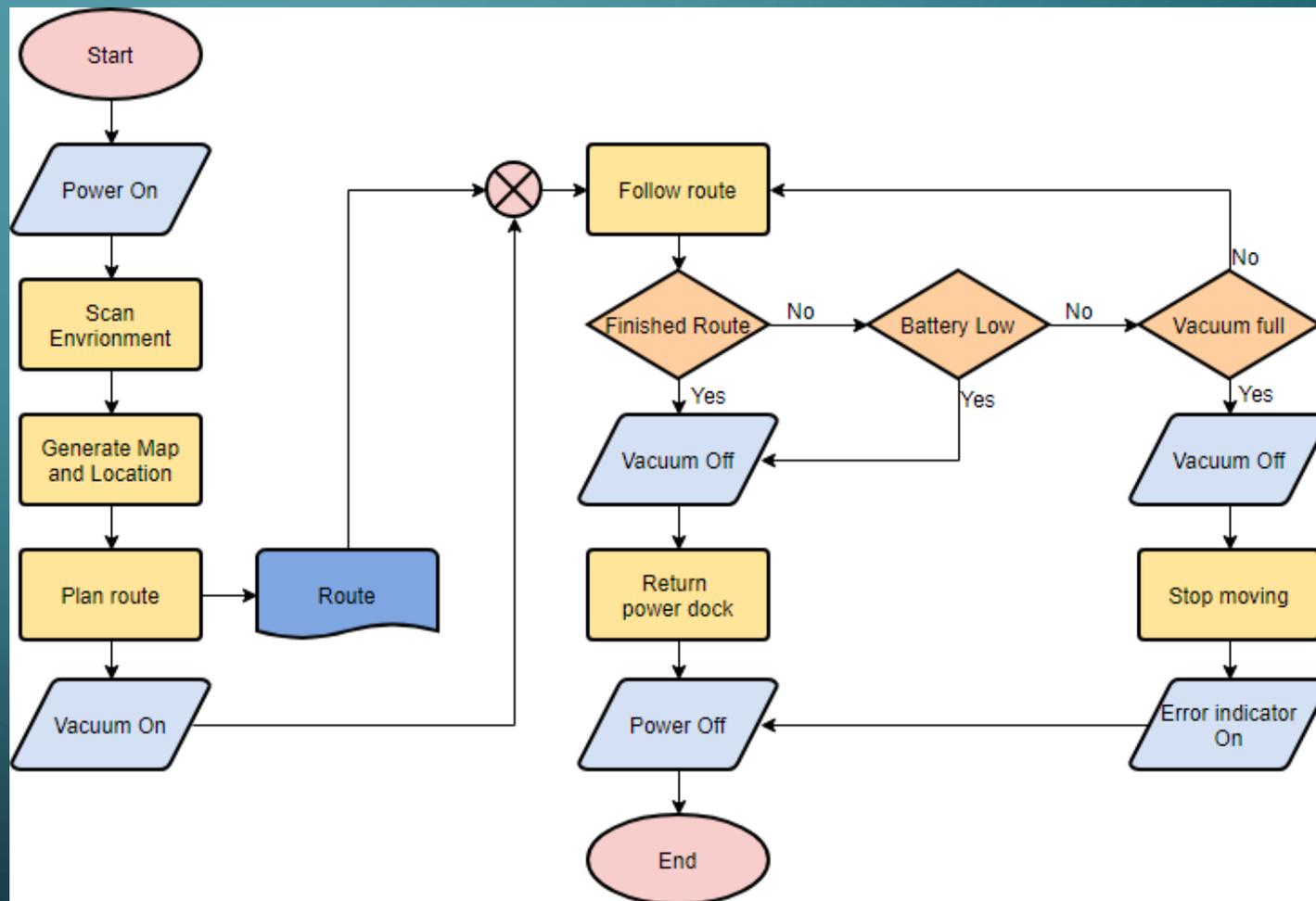
- Алгоритмите са списък от краен брой правила за извършване на действия в определен ред за получаване на решението на всяка задача от даден клас

СВОЙСТВА НА АЛГОРИТМИТЕ

- Определеност
- Масовост
- Дискретност
- Резултатност
- Ефективност

СРЕДСТВА ЗА ОПИСАНИЕ

- Псевдо код
- Блок-схеми(flow charts)
- Програмен език

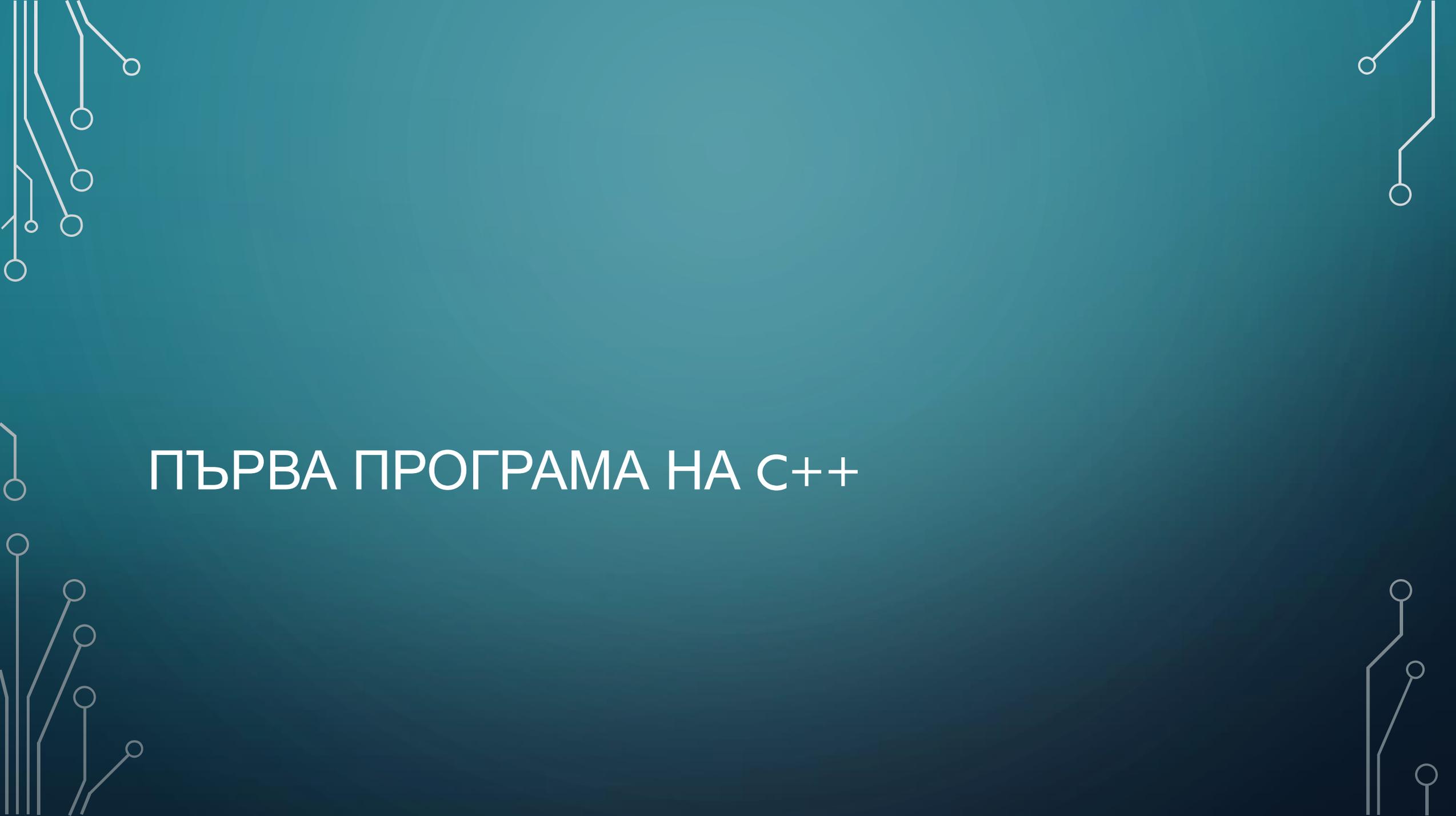


ЕЗИЦИ ЗА ПРОГРАМИРАНЕ

- Машинни езици
- Асемблерни езици
- Процедурно-ориентирани
- Обектно-ориентирани
- други
- Интегрирани среди за разработка –Integrated Development Environment-IDE
- Компилатори-Visual Studio/Code Blocks...

ПРОГРАМЕН ЕЗИК C++

- Разработен през 1985 година от Бьорн Струоструп в AT&T Bell Laboratories
- Разширение на C
- Създаване и използване на абстрактни типове данни
- Обектно-ориентирано програмиране
- Подобрени конструкции
- Гръбнак на други езици и игрови двигатели
- Близък до хардуера



ПЪРВА ПРОГРАМА НА C++

ТИПОВЕ ДАННИ

- Елементи на езика
- Основни типове данни и тяхното представяне в паметта
- Основни операции
- Вход и изход

АЗБУКА НА ЕЗИКА C++

- Главните и малки букви на латинската азбука;
- Цифрите
- Специалните символи
- + - * / = () [] { } | : ; " ' < > , . _ ! @ # \$ % ^ ~

ИДЕНТИФИКАТОР

- Наименование на елемент в програмата
- Поредица от символи
- Букви, цифри и знака за долна черта
- Може да започва само с буква или долна черта
- Валидни идентификатори:
 - `abc`, `ABC`, `minValue1`, `_first_setting`
- Невалидни идентификатори:
 - `1ba`, `ab+1`, `a(1)`

КЛЮЧОВИ ДУМИ

- Вградени в езика идентификатори, които се използват в програмите по стандартен, предварително определен начин
- Ключови думи на C++
 - `int`, `namespace`, `return` и т.н.
- Ключови думи на препроцесора
 - `include`, `undef`, `ifdef` и т.н.
- Ключови думи на компилатора
 - Зависят от конкретния компилатор
 - Например за VS2015: `__finally`, `__event` и т.н.

ОПЕРАТОРИ

В C++ има три групи оператори:

- аритметично-логически оператори
- управляващи оператори
- операторите за управление на динамичната памет

ОПЕРАТОР ЗА ПРИСВОЯВАНЕ

- **<променлива> = <израз>;**
- **като <променлива> и <израз> са от един и същ тип.**
- Пресмята стойността на <израз> и я записва в паметта, именувана

с променливата от лявата страна на знака за присвояване =

ЛИТЕРАЛИ И ПРОМЕНЛИВИ

- Константна стойност на променлива, зададена в кода

```
int i=2;
```

```
char c = 'a';
```

```
bool isReal = true;
```

- `int i = 023` //число в осмична бройна система
- `int i = 0x1F` //число в шеснадесетична бройна система

- Част от паметта, за съхранение на данни което може да съдържа
- различни стойности по време на изпълнение на програмата
- Имат три характеристики
- Тип
- Име
- Стойност
- Името на променливата трябва да е валиден идентификатор
- Всяка променлива трябва да има тип, който се указва при
- дефинирането ѝ

КОНСТАНТИ

- Примери

```
const int MAXINT = 32767;
```

```
const double RI = 2.5 * MAXINT;
```

```
const double PI = 3.14159265;
```

- Предимства
- Програмите стават по-ясни и четливи.
- Лесно (само на едно място) се променят стойностите им (ако се налага).
- Възможността за грешки, възможни при многократното изписване на стойността на константата, намалява.

ОСНОВНИ ТИПОВЕ ДАННИ(1)

- Скаларни
- Булев (bool)
- Символен (char)
- Целочислен (int)
- За числа с плаваща запетая (float, double)
- Изброен (enum)
- Указател (T*)
- Псевдоним (T&)

ОСНОВНИ ТИПОВЕ ДАННИ(2)

Съставни типове

- Масив (`T[]`)
- Символен низ (`char []`)
- Структура (`struct`)
- Клас (`class`)
- Обединение (`union`)

БУЛЕВ(ЛОГИЧЕСКИ) ТИП

- Множество от стойности
- Състои се от два елемента – стойностите `true` (1) и `false` (0)
- `<булева_константа> ::= true | false`
- Основни логически операции
 - `&&` - AND
 - `||` - OR
 - `!` - NOT

<i>X</i>	<i>Y</i>	<i>AND(X,Y)</i>	<i>OR(X,Y)</i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

<i>X</i>	<i>NOT(X)</i>
0	1
1	0

СИМВОЛЕН ТИП

- Променливите от тип `char` са с размер 1 байт
- • Множество от стойности: `[-128; 127]`
- • `unsigned char`: `[0, 255]`
- • Литерали
- • '`<СИМВОЛ>`'
- • '`\<контролен_символ>`',
- • '`\n`' – нов ред
- • '`\t`' – табулация

ЦЕЛОЧИСЛЕН ТИП (INT)

Променливите от тип `int` са с размер от 4 байта

- Множество от стойности: $[-2^{31}; 2^{31}-1]$
- Други целочислени типове
- `short`: $[-2^{15}; 2^{15}-1]$ – 2 байта
- `long`: $[-2^{31}; 2^{31}-1]$ – най-малко 4 байта
- `long long`: $[-2^{63}; 2^{63}-1]$ – най-малко 8 байта
- `unsigned`: $[0; 2^n - 1]$, $n=16,32,64$

ОПЕРАЦИИ С ЦЕЛОЧИСЛЕН ТИП

Аритметични операции

- едноместни операции за знак (+, -)
- двуместни аритметични операции
- $a + b$ (събиране)
- $a - b$ (изваждане)
- $a * b$ (умножение)
- a / b (целочислено деление - частно)
- $a \% b$ (деление по модул - остатък)
- операции за сравнение (предикати)
- $a == b$ (равно)
- $a != b$ (различно)
- $a < b$ (по-малко) • $a > b$ (по-голямо) • $a <= b$ (по-малко или равно) • $a >= b$ (по-голямо или равно)

РЕАЛНИ ЧИСЛА

- В компютъра няма безкрайно количество памет с която да се представят всички реални числа
- Числа с плаваща запетая
- Тип `float`
- Тип `double`

FLOAT

IEEE 754 Floating Point Standard

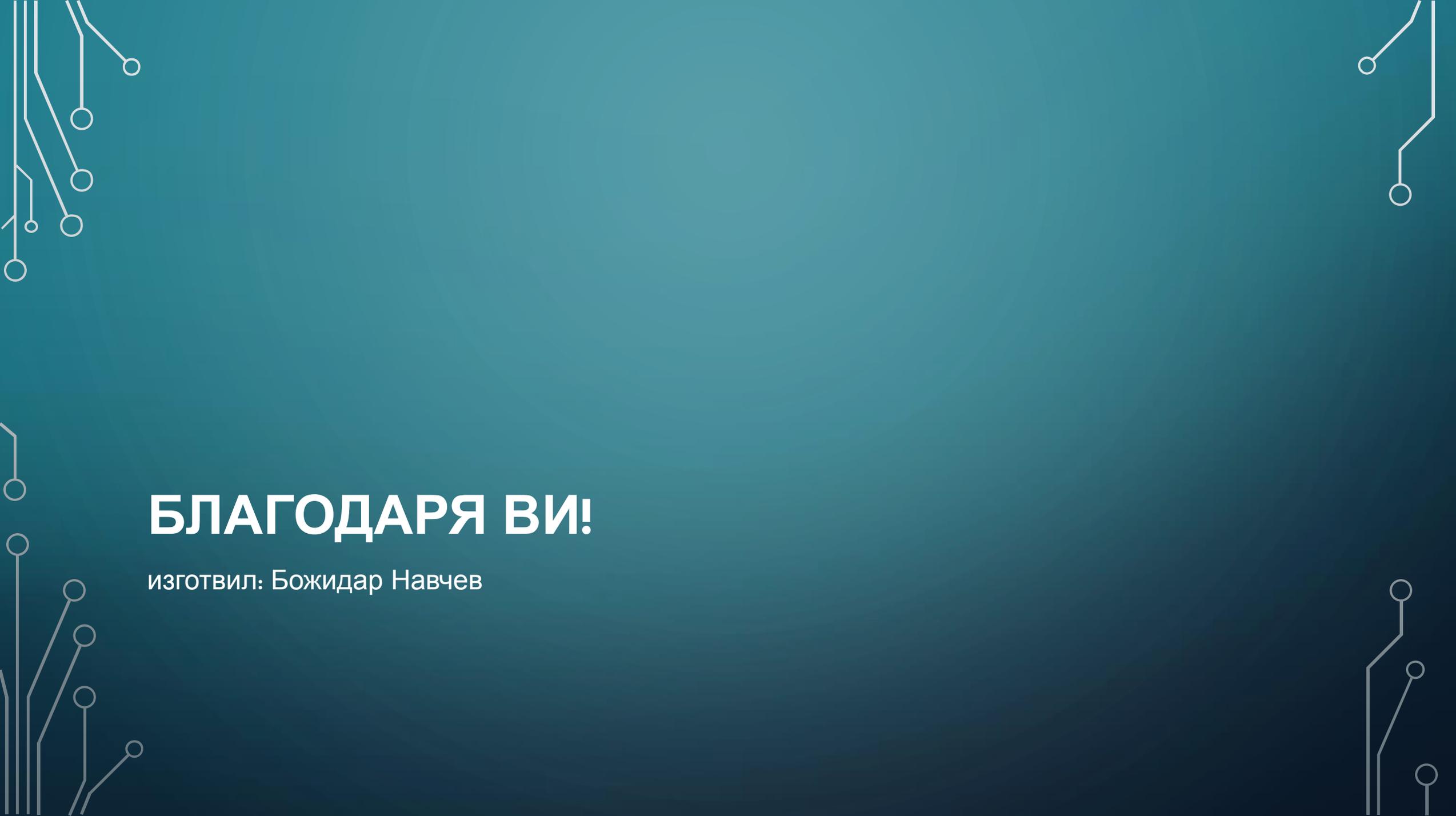


$$\text{number} = (-1)^s * (1.m) * 2^{e-127}$$

- $s: \{0,1\}$ – знак
- $m: [0; 2^{23} - 1]$ - мантиса
- $e: [-128; 127]$ – экспонента
- Тип `double` – 64 бита

ДЕФИНИРАНЕ НА ТИП ИЗБРОЕН

- `enum Weekday{MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY};`
- `enum Name{IVAN=5, PETER=3, MERY=8, SONIA=6, VERA=10};`
- `enum Id{A1, A2, A3, A4=8, A5, A6=10, A7, A8};`
- Ако не са указани стойности, по подразбиране първият идентификатор получава стойност 0, а всеки следващ – стойност с единица по-голяма от стойността на предходния



БЛАГОДАРЯ ВИ!

изготвил: Божидар Навчев